

NOVEL HYBRID ALGORITHM USING CONVOLUTIONAL AUTOENCODER WITH SVM FOR ELECTRICAL IMPEDANCE TOMOGRAPHY AND ULTRASOUND COMPUTED TOMOGRAPHY

Łukasz Maciura¹, Dariusz Wójcik^{1,2}, Tomasz Rymarczyk^{1,2}, Krzysztof Król^{1,2}

¹Research and Development Center, Netrix S.A., Lublin, Poland, ²WSEI University, Lublin, Poland

Abstract: This paper presents a new hybrid algorithm using multiple support vector machines models with a convolutional autoencoder for electrical impedance tomography, and ultrasound computed tomography image reconstruction. The ultimate hybrid solution uses multiple SVM models to convert input measurements to individual autoencoder codes representing a given scene then the decoder part of the autoencoder can reconstruct the scene.

Keywords: convolutional autoencoder, SVM, electrical impedance tomography, ultrasound transmission tomography

NOWY ALGORYTM HYBRYDOWY WYKORZYSTUJĄCY AUTOENKODER KONWOLUCYJNY Z SVM DLA ELEKTRYCZNEJ TOMOGRAFII IMPEDANCYJNEJ I TOMOGRAFII ULTRADŹWIĘKOWEJ

Streszczenie. Artykuł przedstawia nowy hybrydowy algorytm który używa modeli maszyn wektorów nośnych wraz z autoenkoderem konwolucyjnym do rekonstrukcji obrazu z elektrycznej tomografii impedancyjnej oraz ultrasonograficznej tomografii transmisyjnej. Ostateczne rozwiązanie hybrydowe używa wielu modeli SVM do konwersji pomiarów wejściowych do pojedynczych kodów autoenkodera reprezentujących daną scenę a wtedy dekodery wyciągają z autoenkodera może zrekonstruować daną scenę.

Słowa kluczowe: autoencoder konwolucyjny, SVM, elektryczna tomografia impedancyjna, ultradźwiękowa tomografia transmisyjna

Introduction

Ultrasound Transmission Tomography [11, 12, 27] is the process which enables, among others, reconstruction of the scene based on ultrasonic signal measurements. Another [7] paper on ultrasound tomography presents an image reconstruction algorithm based on a conventional neural network (CNN) for two-phase imaging materials. It has also been verified with experimental data. Finally, the paper [4] describes research into applying a dual-domain network for ultrasound tomography.

The EIT reconstruction [16] is based on reconstructing the conductivity from measurements vectors obtained from measurements using electrically conductive electrodes. However, the best results can be obtained using neural networks (especially deep networks). Several works have extensively described the EIT method, among which we can mention works [2, 6, 8, 9, 13, 20, 21, 25].

Several machine-learning methods for EIT reconstruction have been discussed in the literature. In the article [18], the authors explored logistic regression using an elastic net. The paper [16] used neural networks for EIT image reconstruction, where N neural networks trained separately for each output pixel were used to reconstruct image pixel values. Finally, the paper [15] applies convolutional neural network structure in EIT reconstruction.

The multiple ANN reconstruction methods are based on deep and convolutional autoencoders. Paper [10] describes a solution based on EIT reconstruction obtained using a deterministic algorithm (D-Bar) and applies the UNet convolutional model to correct these initial reconstructions.

Another method based on deep convolutional autoencoders is described in [24]. This algorithm uses a deep autoencoder to reconstruct a lung object based on EIT. The method consists of several steps:

1. Minimizing the cost function of the autoencoder with lung images.
2. The fully connected networks are trained on the measurements and outputs of the encoder.
3. Finally, the two networks are combined, where the network from step 2 processes the measurements, and the decoder of the developed autoencoder outputs reconstruction.

There are several numerical methods [5, 14, 17, 19, 22, 23, 24]. The authors have developed an innovative hybrid algorithm for image reconstruction of impedance and ultrasonic transmission tomography. The solution uses the encoder part from the

convolutional autoencoder and multiple Support Vector Machines [3, 26] models to convert EIT or UTT measurements into individual autoencoder codes. In the case of EIT, the synthetic data was used only, containing generated scenes and EIT simulations. Therefore, the autoencoder is trained using generated scenes, and the encoder part is used to encode these scenes. In the final solution, the multiple SVM models are used to convert potential difference vectors from EIT simulations into AE codes representing the corresponding scenes. In the case of UTT, the convolutional autoencoder is trained using augmented real-scene images, and then, the encoder part is used for real-scene encoding; next, multiple SVM models are trained to convert real UTT measurements into codes representing real scenes. The ultimate hybrid algorithm in both cases is constructed using multiple SVM models at the input whose outputs are sent to the decoder part of AE, which reconstructs the image of the scene. Such a hybrid algorithm is particularly useful, especially when there is too small of training data (i.e., where training data comes from real measurements), so impossible to use a pure deep learning solution as in our UTT experiments.

1. EIT synthetic data

The model for the synthetic data in Electrical Impedance Tomography was prepared with 16 electrodes placed outside the area. A finite element mesh was prepared in the model, built with 4717 elements. The EIT training dataset used in this experiment consists of synthetic data. The data generation algorithm generated 50,000 cases containing inclusions (one or two inclusions of elliptical or rectangular shape). An EIT simulation was performed for each case to generate 192 measurements between electrodes.

To generate data, we used an EIT simulation algorithm based on the finite element method with square geometry to obtain potential vectors based on the cases generated with the ellipse and rectangle inclusions. The parameters of the simulation algorithm for generating the training dataset used in the experiments were adjusted to make the synthetic data as close as possible to the real data. Finally, the resulting potential vectors were used to calculate the difference between potential – voltage:

$$X_i = x_i - x_{0i}$$

where x_i – is potential vector for the case with synthetic inclusions, x_{0i} – is potential vector for the case empty tank, X_i – is the voltage vector, i – is the index of potential vectors and voltage elements.



The 50,000 samples generated from reference images (Y) and potential difference vectors (X) were divided into a training dataset (40,000 samples) and a test dataset (10,000 samples).

2. UTT Input data

The data used in this research was obtained in the laboratory under specific conditions. At the bottom of the polypropylene pail, 76 items numbered 1–77 were marked (no. 18 was omitted). The bucket was filled with chlorinated water. Round phantoms in plastic tubes filled with air were placed in marked positions.

An application was written to facilitate data collection. The application was intended to give the positions of the phantoms. The positions of the phantoms were stored in a defined list. In addition, the application controlled the measurements as follows. When the phantom was placed at the indicated position, and the MEASURE button was pressed, the application saved the last measured measurement frame to memory. After the measurement, the application pointed to the next position where the phantom had to be set. This way, the list (measurement matrix) – (position numbers with phantoms) was completed.

Then, thanks to a hand-prepared set of photos (Fig. 2 and based on position numbers, the phantoms' actual positions were reconstructed in binary images. Each image marked with a single inclusion was segmented into a binary image with a circle in that phantom position. The image of the phantom positions in the individual setting variants of several phantoms was obtained by summing up the appropriate segmented position patterns. The resulting image was scaled down to a resolution of 64x64 pixels in the "nearest" interpolation mode.

The active probe of the ultrasonic tomograph performs measurements with a single piezo transducer using the absorption method. This transducer can operate as a transmitter and receiver of the ultrasonic wave. Its resonant frequency is 40 kHz. The PCB allows connecting an external transducer via an SMB socket. The probe has an integrated signal processing circuit and a microcontroller with an integrated A/D converter. Each probe can adjust the gain of the received signal using a programmable digital potentiometer. Measuring the transit time of an ultrasonic wave from one probe to another is achieved by connecting all probes to an additional communication line. When a low state occurs on this line, all probes except the transmitting probe start timing and stop timing when the ultrasound wave is received. Each receiving probe then sends the measurement result to the tomograph controller. Based on the information of which probe was the transmitter and which probe sent the result, the measurement value is stored in the corresponding cell of the measurement matrix. The probes were made to be placed very close to each other. The power supply lines, the communication bus and the interruption lines necessary for correct timing from sending to receiving on the other probes were carried out using RJ12.



Fig. 1. UST active probes on the measuring tank

There were 21 sensors with a frequency of 40 kHz in the bucket area (Fig. 1). The 21x21 size M measurement matrix contains the ToF (Time of Fly) values of the ultrasonic wave between all sensor pairs in microseconds. The diagonal is filled with zeros only to ensure that the value of M_{ij} stands for the value measured between the i -th and j -th sensors and has no effect on the measurements, i.e., the sensor transmitting the wave does not measure the time.

After obtaining the data was split into training (2617 samples) and test (656 samples) datasets.

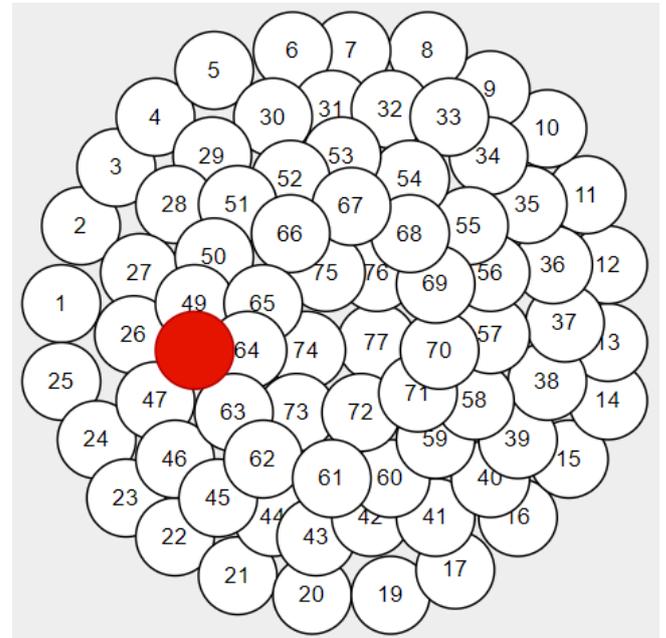


Fig. 2. Sample phantom position during measurements

3. Augmentation of images represented scenes from multimodal tomography (UTT)

Because the collected data from transmission tomography contains a small number of samples (a few thousand), it is too small to train a convolutional autoencoder, so the scene images dataset should be augmented. The image data augmentation is simpler than the augmentation of data tomography measurements. The augmentation of scene images was realized by three transforms performed on images (rotation, scaling and translation) using random parameters. During the augmentation, 25000 images were generated. As a result, 20% of images are empty, and other (80%) images were generated by random affine transformation with maximal rotation change equal to 15 degrees, maximal scale changes equaling 20% and maximal translation equaling 10 pixels. After augmentation, the data was split into a training dataset (containing 20000 images) and a test dataset (containing 5000 images).

4. Training of convolutional autoencoder for EIT data

The convolutional autoencoder for EIT data was trained using synthetic scenes. In table 1, the used convolutional autoencoder model structure details are presented.

The reshape layer (row number 18) in this model makes the reshape operation of input from the previous layer into shape $5 \times 5 \times 1024$ to achieve the proper input shape for the next layer.

The training process was performed using 30 training epochs during six stages, each containing five epochs with different optimizers and settings, presented in table 2.

Table 1. Details of the autoencoder model

id	layer name	neurons number	kernel size	activation
1	Conv2D	64	3×3	relu
2	Conv2D	64	3×3	relu
3	MaxPooling2D	-	2×2	-
4	Conv2D	128	3×3	relu
5	Conv2D	128	3×3	relu
6	MaxPooling2D	-	2×2	-
7	Conv2D	256	3×3	relu
8	Conv2D	256	3×3	relu
9	MaxPooling2D	-	2×2	-
10	Conv2D	512	3×3	relu
11	Conv2D	512	3×3	relu
12	MaxPooling2D	-	2×2	-
13	Conv2D	1024	3×3	relu
14	Conv2D	1024	3×3	relu
15	Flatten	-	-	-
16	Dense	512	-	relu
17	Dense	25600	-	relu
18	Reshape	-	-	-
19	UpSampling2D	-	2×2	-
20	Conv2D	512	3×3	relu
21	Conv2D	512	3×3	relu
22	UpSampling2D	-	2×2	-
23	Conv2D	256	3×3	relu
24	Conv2D	256	3×3	relu
25	UpSampling2D	-	2×2	-
26	Conv2D	128	3×3	relu
27	Conv2D	128	3×3	relu
28	UpSampling2D	-	2×2	-
29	Conv2D	64	3×3	relu
30	Conv2D	64	3×3	relu
31	Conv2D	1	3×3	relu

Table 2. Training stages of autoencoder model

stage	optimizer	learning rate
1	Adam	10 ⁻⁴
2	Adam	10 ⁻⁴
3	Adam	10 ⁻⁵
4	Adam	10 ⁻⁵
5	Adam	10 ⁻⁶
6	Adam	10 ⁻⁷

The 12 samples of pairs containing original scenes (reference images) and scenes images restorations made by autoencoder are presented in figure 3.

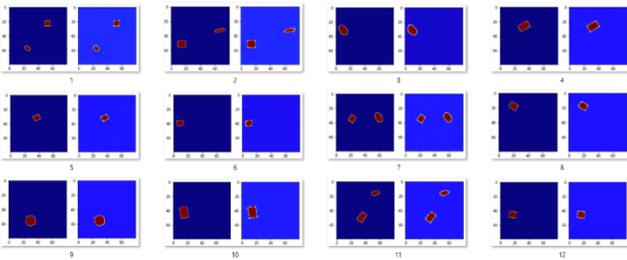


Fig. 3. Sample results of the convolutional autoencoder. In the 12 pairs of samples, there are input scene images (on the left in the pair) and reconstructed input scenes by autoencoder (on the right)

For numerical assessment of results in the form of images, the DICE coefficient [27] was used:

$$DSC = \frac{2TP}{2TP + FP + FN}$$

where:

TP – is the number of true positives (correctly reconstructed pixels), FP – is a number of false positives (pixels reconstructed in wrong places), FN – is some false negatives pixels not reconstructed in the place where the object exists in the real image

The numerical results of pure EIT scenes autoencoder working were the following. After training the autoencoder, the MAE equal 0.0054 and DICE equal 97.7 were achieved on the training dataset, while the MAE equal 0.0058 and DICE equal 96.5 were achieved on the test dataset.

It should be noted that the results presented above are not reconstruction results but are a measure of the autoencoder's ability to reproduce the input image. Nevertheless, this result assures us that the encoding part of the autoencoder can encode the images correctly.

5. Training of convolutional autoencoder for UTT data

The convolutional autoencoder for UTT data was trained on an augmented scene images dataset similar to real scene images. The used model architecture is similar to EIT's, but the Dense layer with an id equal to 17 has 16384 neurons instead of the 25600 ones used for EIT. In addition, the used Reshape layer (id equal 18) has a destination shape equal to 4×4×1024 instead of 5×5×1024 in the previous version of the autoencoder.

The autoencoder was trained using 35 training epochs (7 stages, each containing five epochs) which was presented in table 3.

Table 3. Training stages of autoencoder model

stage	optimizer	learning rate	weight decay
1	Adam	10 ⁻⁴	-
2	Adam	10 ⁻⁴	-
3	Adam	10 ⁻⁴	-
4	Adam	10 ⁻⁵	-
5	Adam	10 ⁻⁵	-
6	AdamWD	10 ⁻⁵	10 ⁻⁸
7	AdamWD	10 ⁻⁵	10 ⁻⁸

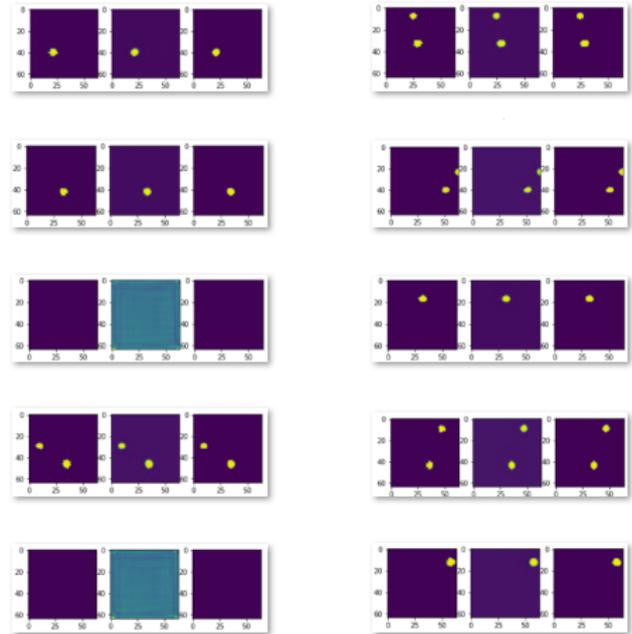


Fig. 4. Samples results of encoding UTT reference images by convolutional autoencoder. In each triplet, on the left is a reference image, in the centre is autoencoder output, and on the right is thresholded autoencoder output

After training, the Mean Absolute Error equal to 0.000826 and DICE equal to 95.37% were achieved on the training dataset, while the MAE equal to 0.0011 and the DICE metric equal to 93.12% were achieved on the test dataset. Then, the encoder part was extracted from the autoencoder pre-trained in the previous step. After extraction, the image scenes from the real dataset were encoded. Finally, the vectors of codes, each containing 512 elements, were generated using the encoder and will be used in the SVM training process.

Figure 3 shows the 10 sample results of EIT reference scene image restoration. Each sample contains a reference image, restored reference image and thresholded restoration result. These samples are generated on the test dataset. The thresholding of outputs is used because empty scenes exist in the dataset, so thresholding was necessary to visualize the output properly.

6. Training of multiple SVM models for EIT and UTT codes represents scenes regression

The final training stage of the ultimate solution uses Support Vector Machines for the regression of codes of scene images based on input from Electrical Impedance Tomography differential potential vectors. The same training stage is performed for Ultrasound Transmission Tomography measurements.

For each i -th code from all 512 available, the separate SVM models are trained for regression one of the code based on input measurements.

In Fig. 5 to Fig. 8, four stages of the algorithm were presented. The input and output of this autoencoder are the same reference image. During training, the autoencoder's reproduction of reference images in the latent space produces a compressed vector representation of the images as a side effect. This representation can be regressed more easily than the original image.

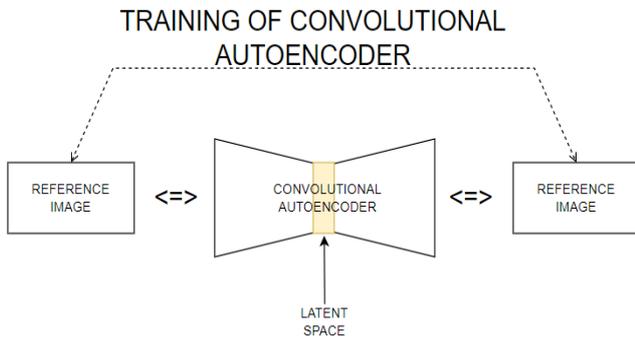


Fig. 5. Training process of the convolutional autoencoder

In Fig. 6, the encoding process of all reference images using the encoder part extracted from the full convolutional autoencoder was presented.

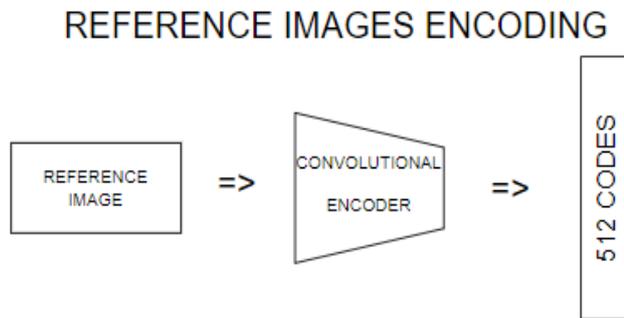


Fig. 6. Reference images encoding (conversion of all reference images into vector)

In Fig. 7, the training process of 512 SVM models, each representing one of 512 codes of autoencoder based on input vector with measures (EIT or UTT), was presented.

In Fig. 8, the reconstruction process was presented. In the beginning, the input vector with measures is used as the same input for each 512 SVM model. Then, each i -th SVM model predicts one i -th value from the autoencoder code vector. Finally, the vector of codes is used as input to extracted decoder part of the convolutional autoencoder to predict the output image (final reconstruction).

During the creation of the solution, the sigma value in the SVM model (SVM parameter) was optimized, and the best result was obtained for an epsilon value equal to 0.01 (in the case of EIT data) and 0.01 (in the case of UTT data). Epsilon is an SVM hyperparameter which is a special margin useful for regression. The model's training tries to fit as many samples as possible in one "street" with as few margin violations. The smaller epsilon values were not tested because of the long training process and bigger memory usage.

TRAINING OF MULTIPLE SVM'S

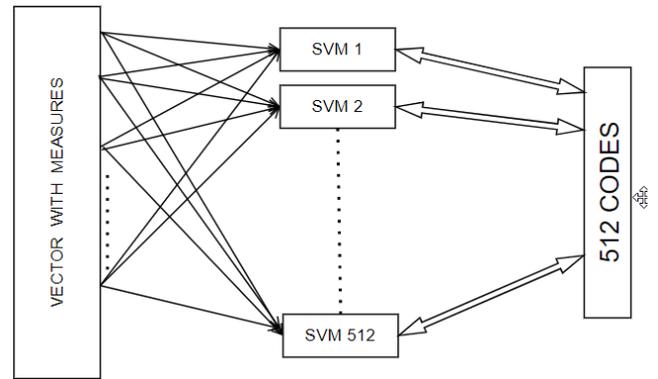


Fig. 7. Training of multiple SVM models to regress 512 codes of vector generated by the encoder

RECONSTRUCTION

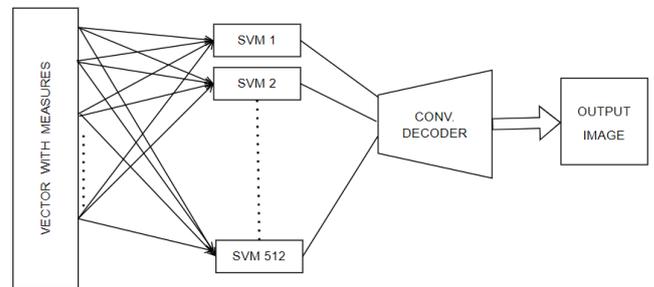


Fig. 8. Whole reconstruction (EIT or UTT) process pipeline

7. Using a decoder with SVM models for transmission tomography image reconstruction for EIT and UTT

The solution for scene image reconstruction has the same form as EIT and UTT. After the pretraining of the autoencoder, the decoder part is extracted. The reconstruction process for EIT and UTT has the following form:

1. In the beginning, the multiple SVM models (trained earlier) are used for individual codes of autoencoder regression based on input measurement (EIT or UTT) where each i -th SVM model can make regression of i -th code in the autoencoder latent space.
2. Obtained vector of codes generated by SVM models is used as an input to the extracted decoder from the pretrained autoencoder, and then the decoder can reconstruct a scene image (for EIT or UTT).
3. The settings of the epsilon parameter are described in paragraphs 9 and 10. The other settings of a single SVM model are the following:
 - a) Input is the vector from EIT or UTT measures.
 - b) Output is one of regressed value represents one of autoencoder code.
 - c) The used kernel type is RBF.
 - d) The gamma parameter is calculated using the following formula:

$$\gamma = \frac{1}{n_f * \sigma_x^2}$$

where:

n_f – is a number of features,
 σ_x^2 – is an input data variance

- e) The tolerance parameter is set to a value of 0.001.
- f) The C parameter is set to 1.
- g) There is no iteration limit.

8. EIT reconstruction results

The numerical results for the EIT reconstruction final solution are presented in table 4.

Table 4. Final results assessments using different SMV models in full hybrid solution (with different epsilon values)

models no.	epsilon	DICE train	DICE test
1	0.2	84.3894%	83.6964%
2	0.1	85.8310%	84.9856%
3	0.05	86.3073%	85.4069%
4	0.01	86.4789%	85.5485%

In figure 9, the 12 pairs of results represent original scenes and reconstructed scenes are presented using the best SVM models (no. 4 in table 2). Next, the table presents the results of four models, identified by the "models no." column. The second column, "epsilon," represents a hyperparameter used in the models. Finally, the "DICE train" and "DICE test" columns show the DICE coefficient, a metric used to evaluate the performance of image reconstruction models for the training and test sets, respectively.

As the epsilon value decreases (i.e. goes from 0.2 to 0.01), the DICE coefficient for both the training and test sets increases. It suggests that as the epsilon value becomes smaller, the model becomes more precise in its reconstruction.

Model 1, with an epsilon of 0.2, has a DICE coefficient of 84.3894% on the training set and 83.6964% on the test set. Model 2, with an epsilon of 0.1, has a DICE coefficient of 85.8310% on the training set and 84.9856% on the test set. Similarly, Model 3, with an epsilon of 0.05 has a DICE coefficient of 86.3073% on the training set and 85.4069% on the test set. Finally, Model 4, with an epsilon of 0.01 has a DICE coefficient of 86.4789% on the training set and 85.5485% on the test set.

It suggests that as the epsilon value decreases, the models become more precise, and the model's performance increases on both training and test sets.

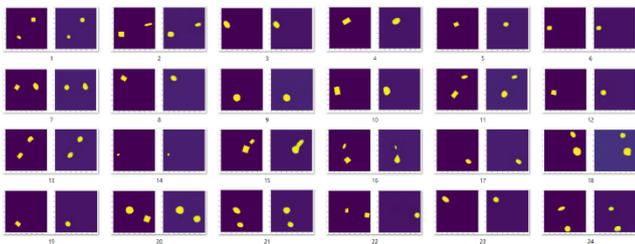


Fig. 9. Sample results of EIT scene reconstructions based on a created hybrid solution. The 12 pairs of results represent original scenes (left images in pairs) and reconstructed scenes (right images in pairs)

9. UTT reconstruction results

The numerical results for the UTT reconstruction final solution were also computed using SVM models trained with epsilon equal to 0.01. The MSE, MAE and DICE metrics are presented in table 5. The table presents the results of a model trained and tested on two different datasets, identified by the "dataset" column. The "MSE" column shows the mean squared error, a metric used to evaluate the performance of regression models, for each dataset. The "MAE" column shows the mean absolute error, another metric used to evaluate the performance of regression models, for each dataset. Finally, the "DICE" column shows the DICE coefficient, a metric used to evaluate the performance of image reconstruction models for each dataset.

The results generally show that the model performed better on the training dataset than on the test dataset. The MSE, MAE and DICE values are generally lower for the training dataset than for the test dataset.

On the training dataset, the MSE is 0.01337, the MAE is 0.01582, and the DICE coefficient is 51.536%. On the test dataset, the MSE is 0.01404, the MAE is 0.01647, and the DICE coefficient is 47.556%.

Table 5. Numerical results of the hybrid model used on UTT data

dataset	MSE	MAE	DICE
training	0.01337	0.01582	51.536%
test	0.01404	0.01647	47.556%

In figure 10, the 8 samples of results of Transmission Tomography reconstruction are presented. Columns 1 and 4 represent the original scene image, columns 2 and 5 represent the reconstructed scene, and columns 3 and 3 represent thresholded results.

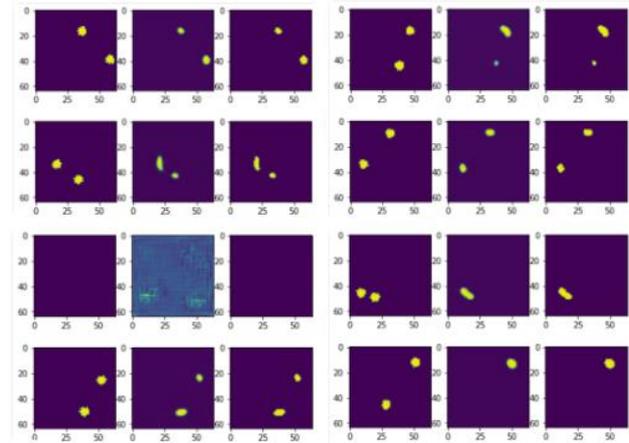


Fig. 10. Sample results of Ultrasound Transmission Tomography reconstruction (columns represent original scenes, reconstructed scenes, thresholded reconstructed scenes)

10. Conclusion

This paper presents a new hybrid algorithm for EIT and UTT scene reconstruction (with all models trained separately). This algorithm consists of multiple SVM models that reconstruct separate codes from the convolutional autoencoder based on EIT or UTT measurements using proper models. Since used convolutional autoencoder could be trained using augmented image data, this hybrid algorithm is very useful for real datasets where data is often limited (as was in our UTT experiments).

Notice that this is our initial research on EIT and UTT scene reconstruction, so the used experimental setup is relatively simple. However, we are planning to use this hybrid method for medical applications. Since convolutional autoencoders have strong possibilities of complex shape reconstruction, we trust that this hybrid solution will also be useful for realistic medical application setups.

The models were implemented in Python with a GPU version of Tensorflow (for convolutional autoencoder) and Scikit – Learn (for SVM libraries).

References

- [1] Aziz Taha A., Hanbury A.: Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. BMC Medical Imaging 15(29), 2015, 1–28.
- [2] Chen B. et al.: Extended Joint Sparsity Reconstruction for Spatial and Temporal ERT Imaging. Sensors 18, 2018, 4014.
- [3] Chen P. H. et al.: A tutorial on v-support vector machines. Applied Stochastic Models in Business and Industry 21, 2005, 111–136.
- [4] Chen Z. et al.: Application of Deep Neural Network to the Reconstruction of Two-Phase Material Imaging by Capacitively Coupled Electrical Resistance Tomography. Electronics 10, 2021, 1058.
- [5] Duraj A., Korzeniewska E., Krawczyk A.: Classification algorithms to identify changes in resistance. Przegląd Elektrotechniczny 91(12), 2015, 82–84.
- [6] Dusek J., Mikulka J.: Measurement-Based Domain Parameter Optimization in Electrical Impedance Tomography Imaging. Sensors 21, 2021, 2507.

- [7] Fan Y. et al.: DDN: dual domain network architecture for non-linear ultrasound transmission tomography reconstruction. Proc. SPIE 11602, 2021, 1160209 [http://doi.org/10.1117/12.2580911].
- [8] Fan Y., Ying L.: Solving electrical impedance tomography with deep learning. Journal of Computational Physics 404, 2020, 109119.
- [9] Fernandez-Fuentes X. et al.: Towards a Fast and Accurate EIT Inverse Problem Solver: A Machine Learning Approach. Electronics 7(12), 2018, 422.
- [10] Hamilton S. J., Hauptmann A.: Deep D – bar: Real time Electrical Impedance Tomography Imaging with Deep Neural Networks. IEEE Trans. Med. Imaging 37(10), 2018, 2367–2377.
- [11] Józefczak A. et al.: Ultrasound transmission tomography-guided heating with nanoparticles. Measurement 197, 2022, [http://doi.org/10.1016/j.measurement.2022.111345].
- [12] Kania K. et al.: Image reconstruction in ultrasound transmission tomography using the Fermat's Principle. Przegląd Elektrotechniczny 96(1), 2020, 186–189.
- [13] Khan T. A., Ling S.H.: Review on Electrical Impedance Tomography: Artificial Intelligence Methods and its Applications. Algorithms 12(5), 2019, 1–18.
- [14] Kłosowski G. et al.: Comparison of Machine Learning Methods for Image Reconstruction Using the LSTM Classifier in Industrial Electrical Tomography. Energies 14(21), 2021, 7269.
- [15] Kłosowski G. et al.: Maintenance of industrial reactors supported by deep learning driven ultrasound tomography. Przegląd Elektrotechniczny 98(4), 2022, 138–147.
- [16] Kłosowski G. et al.: Neural hybrid tomograph for monitoring industrial reactors, Przegląd Elektrotechniczny 96(12), 2020, 190–193.
- [17] Kłosowski G. et al.: Quality Assessment of the Neural Algorithms on the Example of EIT-UST Hybrid Tomography. Sensors 20(11), 2020, 3324.
- [18] Kozłowski E. et al.: Logistic regression in image reconstruction in electrical impedance tomography, Przegląd Elektrotechniczny 96(5), 2020, 95–98.
- [19] Krawczyk A., Korzeniewska E.: Magnetophosphenes—history and contemporary implications. Przegląd Elektrotechniczny 94(1), 2018, 61–64.
- [20] Li X. et al.: An image reconstruction framework based on deep neural network for electrical impedance tomography. IEEE International Conference on Image Processing, Beijing, China, 2017.
- [21] Li X. et al.: A novel deep neural network method for electrical impedance tomography. Transactions of the Institute of Measurement and Control 41(14), 2019, 4035–4049.
- [22] Lukiański M., Wajman R.: The diagnostic of two-phase separation process using digital image segmentation algorithms. Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska 10(3), 2020, 5–8.
- [23] Mosorov V. et al.: Plug Regime Flow Velocity Measurement Problem Based on Correlability Notion and Twin Plane Electrical Capacitance Tomography: Use Case. Sensors 21(6), 2021, 2189 [http://doi.org/10.3390/s21062189].
- [24] Seo J. K. et al.: A Learning – Based Method for Solving III – Posed Nonlinear Inverse Problems: A Simulation Study of Lung EIT, SIAM. Journal on Imaging Sciences 12(3), 2019.
- [25] Szczesny A., Korzeniewska E.: Selection of the method for the earthing resistance measurement. Przegląd Elektrotechniczny 94, 2018, 178–181.
- [26] Yu H., Kim S.: SVM Tutorial: Classification, Regression, and Ranking. Handbook of Natural computing, 2012.
- [27] Zhao W. et al.: Ultrasound transmission tomography image reconstruction with a fully convolutional neural network. Phys Med Biol. 65(23), 2020, 235021, [http://doi.org/10.1088/1361-6560/abb5c3]. PMID: 33245050].

Ph.D Eng. Łukasz Maciura

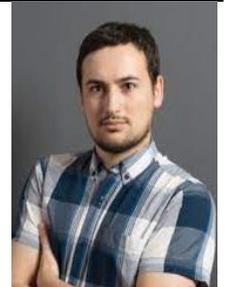
e-mail: lukasz.maciura@netrix.com.pl

Since 2019 he has been working in the industry as a software designer/developer on projects related to computer vision / image processing / deep learning / electrical impedance tomography (currently at Research and Development Center, Netrix S.A.) His research interests include computer vision (especially 3D reconstruction), machine learning (particularly deep learning and time-series recognition), medical informatics and robotics.

<http://orcid.org/0000-0001-8657-3472>**Ph.D. Dariusz Wójcik**

e-mail: dariusz.wojcik@netrix.com.pl

R&D Manager of Algorithmic Systems in Research and Development Center Netrix S.A. The author of the two-fluid numerical code JOANNA focused on the research of the Solar corona. Currently, he is interested in machine learning, neural networks and computer vision and its application to industrial and medical tomography.

<http://orcid.org/0000-0002-4200-3432>**D.Sc. Tomasz Rymarczyk**

e-mail: tomasz.rymarczyk@netrix.com.pl

Director in Research and Development Center Netrix S.A. His research focuses on applying non-invasive imaging techniques, electrical tomography, image reconstruction, numerical modelling, image processing and analysis, process tomography, software engineering, knowledge engineering, artificial intelligence and computer measurement systems.

<http://orcid.org/0000-0002-3524-9151>**M.Sc. Eng. Krzysztof Król**

e-mail: krzysztof.krol@netrix.com.pl

He is an employee of the Research and Development Department at the Research and Development Centre of Netrix S.A. and an assistant at the Higher School of Economics and Innovation in Lublin. He has been associated with computer science for over 14 years. He conducts scientific research in process tomography, IoT and sensor networks. He conducts scientific research in process tomography, IoT and sensor networks.

<http://orcid.org/0000-0002-0114-2794>